## Assignment 1 report template

**Name: Oswaldo Russian**

**CougarNetID: 1299650**

**Name on Kaggle leaderboard: Oswaldo Russian**

**Theory**

Type out the answers to the questions in the table below, taking as many lines as you need. We recommend using the word equation editor for equations. If you would like to use Latex or handwrite the equations, you can paste a screenshot or picture of your work in the corresponding row instead. Handwritten equations which are not legible will receive no points.

|  |  |
| --- | --- |
| A1 | * Linear regression * Perceptron * Logistic regression |
| A2 | Gradient descent (GD) uses a batch update. It needs to cycle through the entire training set to update weights. Stochastic gradient descent (SGD) updates weights based on a partial set of training data. |
| A3 | The first term is the loss associated with predictions of class #1, with label1. The second term is the loss associated with predictions of class #2, with label-1. |
| A4 | The logistic regression loss for a single sample (SGD) is given by  The gradient of this loss w.r.t. ***w*** is given by  \*I am taking the derivative of the sigmoid w.r.t. ***w*** as given.  Then, the SGD update rule for logistic regression is given by |

**Perceptron**

Briefly describe the hyperparameter settings you tried. In particular, you should list the different values for learning rate and number of epochs you tried. You should also mention whether adding a learning rate decay helped and how you implemented this decay. Report the optimal hyperparameter setting you found in the table below. Report your training, validation, and testing accuracy with your optimal hyperparameter setting. If you do the update comparison for extra credit, document your findings and briefly explain to us what you learned. Does the update rule effect the overall accuracies?

**Answer:**

The hyperparameters I tried, and the corresponding accuracy results, are documented in the table below. In general, I implemented learning rate decay in (1/t) form, but did not notice a significant different in accuracy as compared to cases where I didn’t implement learning rate decay.

| **Alpha** | **Num\_Epochs** | **Training Accuracy (%)** | **Validation Accuracy (%)** | **Testing Accuracy (%)** |
| --- | --- | --- | --- | --- |
| 5 | 100 | 36.39 | 25.40 | 27.40 |
| 0.5 | 100 | 38.74 | 28.20 | 29.12 |
| 0.05 | 100 | 37.02 | 28.40 | 27.68 |
| 0.5 | 10 | 33.38 | 28.80 | 27.96 |
| 0.5 | 50 | 35.59 | 26.80 | 26.82 |
| 0.5 | 200 | 39.54 | 25.30 | 28.36 |
| 0.5 | 500 | 40.79 | 27.40 | 28.18 |
| 0.05 | 200 | 40.03 | 26.80 | 28.58 |

|  |  |
| --- | --- |
| Optimal hyperparameters | * alpha = 0.5, * num\_epochs = 100   I implemented learning rate decay in (1/t) form. |
| Training accuracy | 38.74% |
| Validation accuracy | 28.20% |
| Test accuracy | 29.12% |

**Softmax**

Describe the hyperparameter tuning you tried for learning rate, number of epochs, and regularization constant. Report the optimal hyperparameter setting you found in the table below. Also report your training, validation, and testing accuracy with your optimal hyperparameter setting.

**Answer:**

I tried a few of combinations of hyperparameters but did not conduct a systematic parameter optimization. The corresponding accuracy results are documented in the table below.

| **Minibatch Size** | **Alpha** | **Num\_Epochs** | **Regularization Constant** | **Training Accuracy (%)** | **Validation Accuracy (%)** | **Testing Accuracy (%)** |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 0.5 | 100 | 0 | 24.39 | 26.00 | 24.86 |
| 10 | 0.5 | 10 | 0 | 24.40 | 26.00 | 24.82 |
| 100 | 0.05 | 50 | 0 | 24.67 | 25.70 | 25.54 |

|  |  |
| --- | --- |
| Optimal hyperparameters | * Minibatch size = 100, * alpha = 0.05, * num\_epochs = 50, * Regularization constant = 0 |
| Training accuracy | 24.67% |
| Validation accuracy | 25.70% |
| Test accuracy | 25.54% |